

数式処理システム Mathematica を用いた 高校生による数学研究

関西学院高等部 宮寺 良平 (Ryohei Miyadera)
Kwansei Gakuin High School
EM Software 福井 昌則 (Masanori Fukui)
EM Software

1 はじめに

関西学院高等部で行われている数式処理システム Mathematica を用いた数学研究の結果に関して報告する。

今回は近年研究を続けてきて, [1],[2], [3] など で発表 した組み合わせゲームの研究を 発展させることができたので, そのことを報告する. また, 組み合わせゲームに必須な 数学的概念である Grundy 数をうまく説明する方法を考えたので, そのことについても 述べる. 次に最近研究を始めたテーマである「小谷の蟻問題」と「陸上競技のトラック 問題」を紹介する. この2つの問題は高校生レベルの数学を多用するために, 高校数学 のブラッシュアップにもなるので高校生が研究するには適している.

2 チョコレート問題

図 2.1 や 2.2 のような板チョコレートがある. 苦いチョコレートの正方形ブロック (濃 いグレー) と甘いチョコレートの正方形ブロック (グレー) から成り, 線に沿って縦もし くは横一直線にチョコレートを2つにカットして, 苦いチョコレートを含まない方を食 べる. この操作を2人のプレイヤーが繰り返し, 相手に苦いブロックを残した者が勝つ ゲームである. 図 2.1 のような長方形のタイプは, 伝統的な石とりゲームと数学的に同 値なので, 既に必勝法は研究されている. それに対して, 私達は図 2.2 のように右側が 階段状になっているものを研究しており, このタイプのチョコレートを研究したのは私 達が最初である.

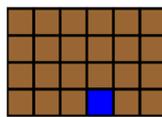


図 2.1

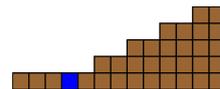


図 2.2

階段状のチョコレート問題を扱うためには, Grundy 数という概念が必要になるが, こ の概念を説明するためにまず棒状のチョコレート問題を扱い, 次にそれを用いて Grundy 数の理論を紹介する.

3 必勝法解析

3.1 棒状チョコレートゲームの必勝法

図 3.1 のような 2 本の棒状のチョコレートがある。棒状のチョコレートは、苦いチョコレートの正方形ブロック (濃いグレー) と甘いチョコレートの正方形ブロック (グレー) から成る。2 人のプレイヤーが交互にチョコレートを切るが、左右いずれかの (どちらを選んでも良い) チョコレートを横線に沿ってカットし、苦い部分を含まない方を食べる。それを交互に繰り返し、相手に 2 つの苦いチョコレートの正方形ブロックを残したプレイヤーが勝つ。ここで先手を A、後手を B とする。図 3.1 の左側のチョコレートは苦いブロックを含めて 5 段あり、右側のチョコレートは苦いブロックを含めて 4 段ある。しかし、プレーにおいては苦いチョコレートのブロックは、食べることができないので、食べることができる甘いチョコレートのブロックだけに注目して、以下の説明において、図 3.1 の左側のチョコレートの高さを 4、右側のチョコレートの高さを 3 と決める。高さは食べられるチョコレートのブロックの個数に対応している。

図 3.1 のゲームの状況では、先手必勝 (先手が最適な戦略をとれば必ず勝つ) である。先手必勝の戦略を説明する。簡単に言えば、先手である A は、自分の番が来る度に、2 つのチョコレートの高さを同じにすると勝つことができる。

まず、先手の A が左側のチョコレートの高さを 3 にすると、2 つのチョコレートの高さは同じ 3 になる。後手である B は、どのようにカットしても 2 つのチョコレートの高さが異なる状態になる。すると、先手の A はまたどちらかをカットすることで、2 つのチョコレートの高さを同じにする。次に後手の B がプレーすると、2 つのチョコレートの高さが異なる状態になる。すると、先手の A はどちらかをカットして、2 つのチョコレートの高さを同じにする。

図 3.2 では、プレーの具体的な 1 つの例を書いている。ここでは、後手の B が右側のチョコレートをカットして高さ 1 とした場合を書いている。次に先手の A は、高さが高い方のチョコレートをカットして、2 つのチョコレートの高さが同じ 1 になるようにする。それを繰り返すと、先手の A は左右のチョコレートがいずれも苦い部分だけとなる状態に持っていきけるため、後手の B が負けとなる。

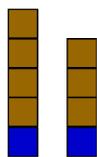


図 3.1

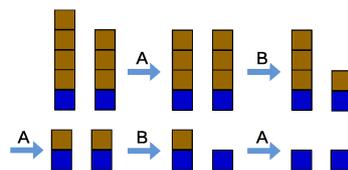


図 3.2

次に、図 3.1 のチョコレートの代わりに、図 3.3 のチョコレートでゲームを考える。その場合、どのようにプレーすればよいだろうか。ここで、左側のチョコレートは、水平方向のカットしかできないが、この右側のチョコレートは図 3.4 にあるように、水平方向と垂直方向のカットが可能である。

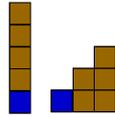


図 3.3

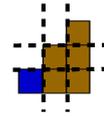


図 3.4

図 3.3 の左側のチョコレートは、高さが4である。一方、図 3.3 の右側のチョコレートは、どのようなチョコレートとして考えたらよいのだろうか。

3.2 階段状チョコレートの Grundy 数

ここからしばらくの間は左側の棒状のチョコレートのことは考えずに、図 3.3 の右側のチョコレートのみについて考える。すなわち、図 3.5 のチョコレートについて考える。



図 3.5

このチョコレートを扱うためには、Grundy 数というものが必要になる。Grundy 数とは、まず図 3.6 の Grundy 数を 0 と定義し、そのチョコレートが一手で進める行き先に含まれない最小の非負整数として再帰的に定義される数である。このような定義だけでは分かりにくいので、以下で具体的な計算例を書く。まず、図 3.6 のチョコレートの Grundy 数は 0 である。

次に図 3.7 のチョコレートを考える。図 3.7 のチョコレートを切ると図 3.6 になる。このことは、図 3.8 に示してある。図 3.6 の Grundy 数が 0 なので、Grundy 数の定義より、図 3.7 の Grundy 数は 1 となる。



図 3.6



図 3.7



図 3.8

次に図 3.9 のチョコレートを考える。図 3.9 のチョコレートを切ると、1 列切るか、あるいは 2 列切るかによって図 3.7 のチョコレートか、あるいは図 3.6 のチョコレートになる。このことは、図 3.10 に示してある。それぞれの Grundy 数が 1 と 0 なので、図 3.9 の Grundy 数は 2 となる。



図 3.9

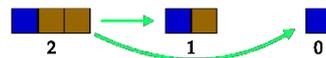


図 3.10

図 3.11 のチョコレートを水平にカットすると図 3.7 のチョコレートとなり、垂直にカットすると図 3.6 のチョコレートになる。このことは図 3.12 に示してある。それぞれの Grundy 数が 1 と 0 なので、図 3.11 の Grundy 数は 2 となる。

よって、図 3.5 のチョコレートは、図 3.13 のようなカットによって 4 つのチョコレートになる。それぞれのチョコレートの Grundy 数が 1,2,2,0 なので、図 3.5 の Grundy 数は 3 であることがわかり、これがチョコレートの高さに対応する。つまり、図 3.3 のゲームは、図 3.13 を用いると、左右のチョコレートの数を揃えるようにプレーすると勝てるという考え方をそのまま用いることができる。

実際に図 3.3 のゲームを具体的にプレーしてみる。最初の状態では、左側のチョコレートの高さは 4、右側の高さは 3 であるから、先手である A は左側のチョコレートの高さを 3 にすればよい。後手である B が図 3.11 のチョコレートになるようにカットしたとする。その Grundy 数は 2 であるから、先手の A は左側のチョコレートの高さを 2 にすればよい。次に後手の B が図 3.7 のチョコレートになるようにカットしたとすれば、その Grundy 数は 1 である。よって先手の A は左側のチョコレートの高さを 1 にすればよい。次に後手の B は左右いずれかのチョコレートを高さ 0 にすることしかできず、後手の B が負けとなる。



図 3.11

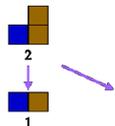


図 3.12

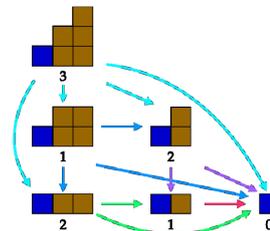


図 3.13

3.3 Grundy 数がニム和に等しいチョコレートゲーム

この問題は関西学院高等部の中屋悠資君と、平林直樹君が研究したものである。私たちは、Grundy 数はニム和と一致するチョコレートゲームについて研究している。そして、以下のような Mathematica の code を用いる。

```
Clear[allcases, move, Mex, Gr, gote, k, ss, res, pposition, ff, bl,
res2]; st = 8; hs = 1; st2 = st;
res[x_] := Floor[x/2] + (st2); res2[x_] := Floor[x/2] + st;
ss = 20; a1 = Flatten[Table[{a, b}, {a, 0, ss}, {b, 0, ss}], 1];
allcases = Select[a1, res#[[1]] >= #[[2]] && res2#[[2]] >= #[[1]] &];
move[z_] := Block[{p}, p = z;
Union[Table[{t1, Min[res[t1], p[[2]]]}, {t1, 0, p[[1]] - 1}],
Table[{Min[p[[1]], res2[t2]}, t2}, {t2, 0, p[[2]] - 1}]];
Mex[L_] := Min[Complement[Range[0, Length[L]], L]];
Gr[pos_] := Gr[pos] = Mex[Map[Gr, move[pos]]];
pposition = Select[allcases, Gr[#] == 0 &]; gg = 0.75;
ff[x_] := If[{x[[1]], x[[2]]} == {-1, -1}, "",
If[x[[2]] == -1, x[[1]], If[x[[1]] == -1, x[[2]],
If[res[x[[1]]] >= x[[2]] && res2[x[[2]]] >= x[[1]],
Gr[{x[[1]], x[[2]]} - hs*BitXor[x[[1]], x[[2]], ""]]];
bl = Select[Flatten[Table[{n + 2, m + 2}, {n, 0, ss}, {m, 0, res[ss]}],
1], ! (res[(#[[1]] - 2)] >= (#[[2]] - 2) &&
res2[(#[[2]] - 2)] >= (#[[1]] - 2)) &];
b2 = Table[{1, t + 2}, {t, 0, res[ss]}];
b3 = Table[{t + 2, 1}, {t, 0, ss}]; a1 = {{2, 2}};
```

```
Print[Grid[Table[ff[{n, m}], {n, -1, ss}, {m, -1, res[ss]}],
Frame -> All, Background -> {None, None, Join[
Table[b1[{{s}} -> GrayLevel[0.87], {s, 1, Length[b1]}],
Table[b2[{{s}} -> GrayLevel[gg], {s, 1, Length[b2]}],
Table[b3[{{s}} -> GrayLevel[gg], {s, 1, Length[b3]}]}]}]]
```

変数 $hs=0$ とすると、表 3.14 ができる。これは、この形のチョコレートの Grundy 数の表である。 $hs=1$ とすると、表 3.15 ができる。これは各 Grundy 数 $G(\{x, y\})$ から $x \oplus y$ を引いた値である。 \oplus はニム和 (排他的論理和) の記号である。表 3.15 ではチョコレートの先端だけが 0 でない。このことから、この形のチョコレートは先端以外で $G(\{x, y\})=x \oplus y$ となる。このような方法で、Grundy 数 $G(\{x, y\})=x \oplus y$ となるようなチョコレートを探している。なお、表 3.14, 表 3.15 のような形のチョコレートについては、現在研究中である。

0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	1	2	3	4	5	6	7	8									
1	1	0	3	2	5	4	7	6	9									
2	2	3	0	1	6	7	4	5	10	11								
3	3	2	1	0	7	6	5	4	11	10								
4	4	5	6	7	0	1	2	3	12	13	14							
5	5	4	7	6	11	0	3	2	13	12	15							
6	6	7	4	5	2	3	0	1	14	15	12	13						
7	7	6	5	4	3	2	1	0	15	14	13	12						
8	8	9	10	11	12	13	14	15	0	1	2	3	4					
9		11	10	13	12	15	14	1	0	3	2	5						
10			14	15	12	13	2	3	0	1	6	7						
11				13	12	3	2	1	0	7	6							
12					4	5	6	7	0	1	2							
13						7	6	1	0	3								
14							2	3	0	1								
15								1	0									
16																16		
17																		
18																		
19																		
20																		

表 3.14 Grundy 数の表

0	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
0	0	0	0	0	0	0	0	0	0										
1	0	0	0	0	0	0	0	0	0										
2	0	0	0	0	0	0	0	0	0										
3	0	0	0	0	0	0	0	0	0										
4	0	0	0	0	0	0	0	0	0	0									
5	0	0	0	0	0	0	0	0	0	0									
6	0	0	0	0	0	0	0	0	0	0	0								
7	0	0	0	0	0	0	0	0	0	0	0	0							
8	0	0	0	0	0	0	0	0	0	0	0	0	0						
9		0	0	0	0	0	0	0	0	0	0	0							
10			0	0	0	0	0	0	0	0	0	0	0						
11				0	0	0	0	0	0	0	0	0	0						
12					0	0	0	0	0	0	0	0	0						
13						0	0	0	0	0	0	0	0						
14							0	0	0	0	0	0	0						
15								0	0	0	0	0	0	0					
16									0	0	0	0	0	0	0				
17										0	0	0	0	0	0	0			
18											0	0	0	0	0	0	0		
19												0	0	0	0	0	0	0	
20													0	0	0	0	0	0	0

表 3.15 Grundy 数からニム和を引いた表

4 小谷の蟻問題

この問題は関西学院高等部の戸國友貴君が主に研究したものである。

小谷の蟻問題では、図 4.1 の I のように一辺 1 の立方体を二つ重ねて作る直方体の表面を蟻が動くとき、点 O から最も遠い場所を求める。最も遠いとは、最短のコースを使うという条件で最も遠いということである。小谷の問題は文献 [5] に詳しい解説があり、また web 上では Kotani's ant という問題として紹介されている。

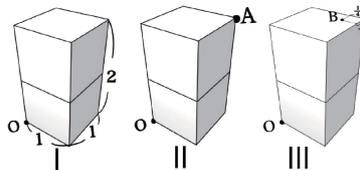


図 4.1

直感的には一番遠い場所は下の図 4.1 の II にある頂点 A だと思える。しかし多くの人にとって意外な事には、一番遠い場所は、上面の内側にあり、図 4.1 の III のように反対側の頂点から縦横それぞれ $\frac{1}{4}$ の長さだけ入った点 B である。

5 四角錐台の場合

小谷の蟻問題に関しては、円柱の場合や任意の直方体等が研究されているが、図 5.1 の *I* のような四角錐台の場合は、先行研究が見つからなかったので、研究することにした。点 O から最短距離で移動した場合に最も遠い点を求める。この場合には逆三角関数と無理関数が入ってくる。小谷の問題のように数学的手法だけで行なうことは無理だと判断して、数学的な式を作ってから計算機による数値計算で解を探した。

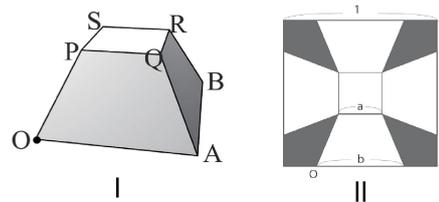


図 5.1

四角錐台は図 5.1 の *II* のように一辺の長さが 1 である正方形から四隅を切り取るとする。ここで底は別の紙で覆うとする。なお使う一辺が 1 という制限をつけた理由については、いろんなタイプの四角錐台を比べるためには、何かの値を一定にしておいた方がよいからである。

最短距離で移動した場合に、最も遠い点は点 O において図形を左右対称に分ける場合の対称面（垂直な）と四角錐台の表面が交差する部分に存在することを証明する。

点 O を通る面によって四角錐台を左右対称に分けて、四角錐台の右側と左側と呼ぶことにする。右側から近づくとか、左側から近づくという表現については、図 5.2 を見ていただきたい。

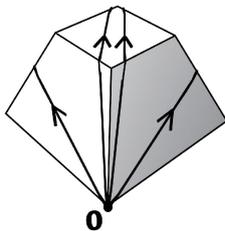


図 5.2

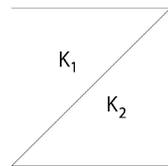


図 5.3



図 5.4

次の補題 5.1 と定理 5.1 の証明は略するが、文献 [5] にある証明を精密化するだけで証明することができる。

補題 5.1. 最短距離で移動した場合に、最も遠い点は、四角錐台の右側から近づいても左側から近づいても最短距離は同じである。

定理 5.1. 最短距離で移動した場合に、最も遠い点が点 O において図形を左右対称に分ける場合の対称面と四角錐台の表面が交わるところに存在する。

なお、ここでは対称面以外に O からの最短距離が対称面にある最も遠い点の最短距離に等しいような点が存在する可能性は否定していない。

定理 5.1 により、最短距離で移動した場合に、最も遠い点は少なくとも 1 つは対称面と四角錐台の表面が交わっているところに存在するので、この部分に存在する各点について最短距離を求め、その最短距離が最大になる点を探す。点の場所については、上面、側面、底面があり、それらを順番に調べれば良い。

以下においていろんな展開図が現れるが、その中で添字のついた点は、図 5.1 の I や図 5.2 において添字のついていない点に対応する。例えば Q_1, Q_2, \dots などはずべて図 5.1 の I や図 5.2 においての点 Q のことであるが、展開図を作る時に決めた平面座標において点 Q を考えるために添字をつけて考えている。

5.1 上面にある点の場合

まず、四角錐台の上面（地面に平行な面）にある点について、 O からの最短距離を調べる。この場合対角線 PR 上で探すことになる。この場合は、図 5.5、図 5.6、図 5.7 の場合に分かれる。

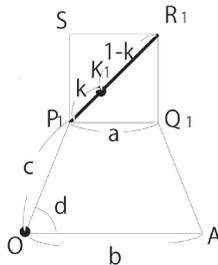


図 5.5

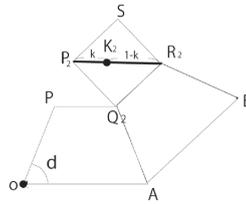


図 5.6

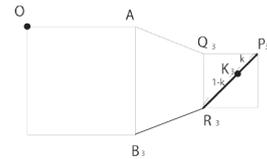


図 5.7

図 5.5 の場合は、点 O を出発してから側面の台形 OAP_1Q_1 を通って、上面の正方形 $P_1Q_1R_1S$ に入って、その内部にある点 K_1 へ到達する。点 $O = (0, 0)$ として各点に座標を決めるが、展開図の種類によって四角錐台では同じ点であるものが違った座標で表されることがある。そのような場合は添字をつける。図 5.5 では P_1, Q_1, R_1 などで表されている。この場合点 A の座標は関係ないので、この場合は A には添字をつけていない。

図 5.1 の II により、点 $P_1 = (\frac{b-a}{2}, \frac{1-a}{2})$, 点 $Q_1 = (\frac{a+b}{2}, \frac{1-a}{2})$, 点 $R_1 = (\frac{a+b}{2}, \frac{1+a}{2})$ となる。最短距離を測定する点 K_1 は P_1R_1 を $k : (1-k)$ に内分すると仮定すれば、

$$K_1 = kR_1 + (1-k)P_1 \quad (1)$$

となる。ここで (1) において右辺はベクトル（位置ベクトル）のスカラー倍と和と考える。この場合は、点 O から点 K_1 までの距離は (1) によって計算される。

次に図 5.6 の場合では、点 O から、側面の台形 OAP_2Q_2 を通って、次にその隣の側面の台形 ABR_2Q_2 を通ってから上面の正方形 $P_2Q_2R_2S$ に入って、その内部にある点 K へ到達する。四角錐台の形によってはこのコースが最短になる可能性がある。ここでも、使う文字に関しては添字をつけて図 5.5 の場合と区別している。

辺 $c = \sqrt{(\frac{b-a}{2})^2 + (\frac{1-a}{2})^2}$, $\cos d = \frac{b-a}{2c}$ により、角 $d = \cos^{-1} \frac{b-a}{2c}$ となる。まず、 $Q_2 = Q_1 = (\frac{a+b}{2}, \frac{1-a}{2})$ である。点 $R_2 = Q_2 + (a \cos(\pi - 2d), a \sin(\pi - 2d))$, 点 $P_2 = Q_2 +$

$(-a \cos(2d - \frac{\pi}{2}), a \sin(2d - \frac{\pi}{2}))$. K_2 は P_2R_2 を $k : (1 - k)$ に内分すると考えると, 点 K_2 は次のようになる.

$$K_2 = kR_2 + (1 - k)P_2. \quad (2)$$

このとき点 O からの最短距離は次の2つの場合があり得る.

Case1 線分 OQ_2 の傾きが線分 OK_2 の傾きよりも大きい場合. この場合は, 最短距離として線分 OK_2 の長さ (2) を使う.

Case2 線分 OQ_2 の傾きが線分 OK_2 の傾きよりも小さい場合は, 2つの側面の台形と上面の正方形だけを通るという条件で考えると, 最短距離は Q_2 の部分で曲がらないといけないが, これが最短距離にならないことは明らかである.

図5.7の場合では, 点 O から, 底面の正方形 OAB_3C を通って, 次にその隣の側面の台形 $AB_3R_3Q_3$ を通ってから上面の正方形 $Q_3R_3SP_3$ に入って, その内部にある点 K_3 へ到達する点 $Q_3 = A + (\frac{1-a}{2}, \frac{a-b}{2})$ であるから, 点 $R_3 = (\frac{2b-a+1}{2}, \frac{-a-b}{2})$, 点 $P_3 = (\frac{2b+a+1}{2}, \frac{a-b}{2})$ となり, K_3 が P_3R_3 を $k : (1 - k)$ に内分するとすれば, 次のようになる.

$$K_3 = kR_3 + (1 - k)P_3. \quad (3)$$

このとき点 O からの最短距離は次の2つの場合があり得る.

Case1 線分 OQ_3 の傾きが線分 OK_3 の傾きよりも大きい場合. この場合は, 最短距離として線分 OK_3 の長さ (3) を使う.

Case2 線分 OQ_3 の傾きが線分 OK_3 の傾きよりも小さい場合は, 底面の正方形から側面の台形を経て上面の正方形という道順を守ると, 最短距離は Q_3 の部分で曲がらないといけないが, これは最短距離にならない.

このやり方で, 次には側面に点がある場合, 底面にある場合などを調べて, 式を作る. そしてそれらの式に数値を代入することで, 最短距離を考えたときに, 一番遠い点を計算することができる.

6 陸上競技のトラック問題

この問題は関西学院高等部の戸國友貴君が主に研究したものである.

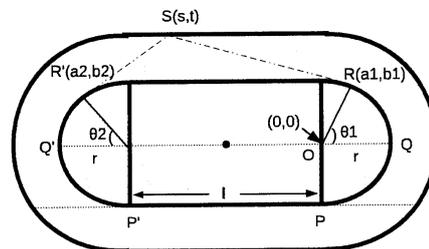


図6.1 陸上競技トラック

小谷の問題は3次元の図形で考えるが, 小谷の問題から新しい問題を考えるときに, 次元を下げた. 小谷の問題の一番の特徴は, 最も遠いという概念の難しさである. 最も遠いということが簡単には求めることができないような例を2次元で探してみても, 陸上

競技場のトラック内だけを動けて、トラックの外にもフィールドの中にも入れないという条件で考えた。これは競技としては、1500m, 5000m, 10000m 走において走ることができる場所に相当する。問題を定式化すると次のようになる。陸上競技のトラック(以下、トラックと略記)があり、その外側にトラックから幅が同じ外周があるとする。このトラックは、長方形の左右に半円を付けたものであり、陸上競技で一般的に用いられている。

トラック上のある点 P から、トラックの内側に入らないように最短経路で進みながら、到達することが出来る一番遠い点はどこか？

図 6.1 のように座標および点を決める。点 Q から点 S へ最短距離で行くには、 P から円周に沿ってある点 R まで行き、そこでの接線を通して点 S へ行くとする。 $PQRS$ の距離 L_1 を求める。なお、 $s < 0, t > 0$ と仮定しても一般性を失わない。

$$\widehat{PQ} = 2\pi r \times \frac{1}{4} = \frac{\pi r}{2} \quad (4)$$

$$\widehat{QR} = r\theta_1 \quad (0 \leq \theta_1 \leq \frac{\pi}{2}) \quad (5)$$

$$\sin \theta_1 = \frac{b_1}{r} \text{ より, } \theta_1 = \sin^{-1} \frac{b_1}{r} \text{ だから,}$$

$$\widehat{QR} = r \sin^{-1} \frac{b_1}{r} \quad (6)$$

$$RS = \sqrt{(s - a_1)^2 + (t - b_1)^2} \quad (7)$$

$$x^2 + y^2 = r^2 \quad (8)$$

(8) の円において、点 R における接線の方程式は、

$$a_1x + b_1y = r^2 \quad (9)$$

であり、この接線が点 S を通るから、

$$a_1 \cdot s + b_1 \cdot t = r^2 \quad (10)$$

が成り立つ。(a_1, b_1) は、(8) の円上の点なので、

$$a_1^2 + b_1^2 = r^2 \quad (11)$$

を満たす。

式(9), (11)を使い、点 R は求められる接点のうちで、右側なので次のようになる。

$$a_1 = \frac{r^2s + \sqrt{r^2t^2(-r^2 + s^2 + t^2)}}{s^2 + t^2} \quad (12)$$

$$b_1 = \frac{r^2t^2 - s\sqrt{r^2t^2(-r^2 + s^2 + t^2)}}{t(s^2 + t^2)} \quad (13)$$

従って、 $PQRS$ の距離 L_1 は、

$$L_1 = \widehat{PQ} + \widehat{QR} + RS = \frac{\pi r}{2} + r \sin^{-1} \frac{b_1}{r} + \sqrt{(s - a_1)^2 + (t - b_1)^2}. \quad (14)$$

次に $PP'Q'R'S$ の距離 L_2 を求める. L_1 の場合とほぼ同じやり方で求めることができる.

$$L_2 = PP' + P'Q' + Q'R' + R'S = l + \frac{\pi r}{2} + r \sin^{-1} \frac{b_2}{r} + \sqrt{(s - a_2)^2 + (t - b_2)^2} \quad (15)$$

補題 5.1 はこの問題においても成り立つので, L_1 と L_2 が等しいとき, 距離が一番遠い場所となる.

7 計算機による陸上競技トラック問題の計算

次にこれまでの理論を使った Mathematica の式を書く. ここでは簡単にするために $r = 1, l = 2$ として計算している.

```
f1[s_, t_] := Block[{x1, y1}, x1 =  $\frac{s + \sqrt{-t^2 + s^2 t^2 + t^4}}{s^2 + t^2}$ ;
y1 =  $\frac{1 - \frac{s^2}{s^2 + t^2} - \frac{s \sqrt{t^2(-1 + s^2 + t^2)}}{s^2 + t^2}}{t}$ ;
(ArcCos[x1] + Pi/2) + EuclideanDistance[{s, t}, {x1, y1}]
f2[s_, t_] := Block[{x2, y2}, x2 =  $\frac{-6 - 7s - 2s^2 - 2t^2 - \sqrt{3t^2 + 4st^2 + s^2 t^2 + t^4}}{4 + 4s + s^2 + t^2}$ ;
y2 =  $\frac{1}{t}(-3 - 2s + \frac{12}{4 + 4s + s^2 + t^2} + \frac{20s}{4 + 4s + s^2 + t^2} + \frac{11s^2}{4 + 4s + s^2 + t^2} + \frac{2s^3}{4 + 4s + s^2 + t^2} + \frac{4t^2}{4 + 4s + s^2 + t^2} + \frac{2st^2}{4 + 4s + s^2 + t^2} + \frac{2\sqrt{3t^2 + 4st^2 + s^2 t^2 + t^4}}{4 + 4s + s^2 + t^2} + \frac{s\sqrt{3t^2 + 4st^2 + s^2 t^2 + t^4}}{4 + 4s + s^2 + t^2})$ ;
(ArcCos[-2 - x2] + Pi/2) +
EuclideanDistance[{s, t}, {x2, y2}] + 2];
```

上の関数を使って, 右回りと左周りが等しい位置を探してプロットする.

```
data = Table[{FindRoot[f1[s, t] == f2[s, t], {s, -2}][[1, 2]], t}, {t, 1, 2, 0.01}];
sq = Join[Table[{0, tt}, {tt, -1, 1, 0.01}], Table[{-2, tt}, {tt, -1, 1, 0.01}],
Table[{ss, 1}, {ss, -2, 0, 0.01}], Table[{Cos[dd], Sin[dd]}, {dd, -Pi/2, Pi/2, 0.01}],
Table[{Cos[dd] - 2, Sin[dd]}, {dd, Pi/2, 3Pi/2, 0.01}], Table[{ss, -1}, {ss, -2, 0, 0.01}]];
ListPlot[{sq, data}, PlotStyle -> {GrayLevel[0.6], Black}, PlotRange -> All,
AspectRatio -> Automatic]
```

図 7.1 において, 左上から上方へ上がっている曲線が左周りと右周りの最短距離が同じ場所である. 従って, 陸上トラックの外周とこの曲線が交わる場所が一番遠いところである.

補足 7.1. 陸上トラックでの最も遠い位置というのは, 計算は複雑になるが, 問題そのものとしてはそれほど難しくない. しかしこの問題をもう少し一般化して, トラック内部で一番遠い 2 点を求めるとするともっと難しくなる.

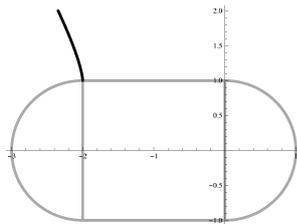


図 7.1

参考文献

- [1] 宮寺 良平, 井上 泰志, 小笠 航, 中村 駿佑, 石取りゲームの変種であるチョコレートゲーム, 情報処理学会論文誌, 53(6) pp. 1582-1591.
- [2] Miyadera,M., Nakamura,S., Okada,Y., Hanafusa,R. and Ishikawa,T.:Chocolate Games -How High School Students Discovered New Formulas Using Mathematica-, Mathematica Journal, Volume 15, (2013).
- [3] Nakamura,S. and Miyadera, R.: Impartial Chocolate Bar Games, Integers 15 (2015).
- [4] 福井 昌則, 宮寺 良平, 数式処理システムと組み合わせゲーム論, 京都大学数理解析研究所, No.1955, pp.81-90, 2015.
- [5] 小谷 善行, 最も遠い場所はどこか?, 図形パズル アリの問題, Newton 別冊 数学パズル チャレンジ 超問 120, p20,21